A Spectrum of IV&V Modeling Techniques

Definition of Metrics-Collection Methods[1]

1. University of Minnesota (UMN) Modeling Languages and Test Engines

Detailed information concerning UMN's RSML^{-e} language, related tools, and past case studies can be found in the "Case Study" report associated with this project, or the following documents: "Collect Models from UMN Clients" (umn_models.pdf), "Definition of UMN Languages: RSML^{-e}" (umn_languages.pdf), and "Definition of UMN Test Engines" (umn_test_engines.pdf).

2. West Virginia University (WVU), NASA IV&V Modeling Languages and Test Engines

Detailed descriptions of WVU / NASA languages and models can also be found in the "Case Study" report, or: "Definition of WVU Languages" (wvu_languages.pdf) and "A Model-Base Approach to Reactive, Self-Configuring Systems" (livingstone.pdf). WVU / NASA's partial random search test engine, to be evaluated in this project, is described in "Definition of WVU Test Engines" (wvu_test_engines.pdf). For more information see also "An Alternative to Model Checking: Verification by Random Search of AND-OR Graphs Representing Finite-State Models" (alternative.pdf).

3. Evaluation of New Test Engines

WVU's new automatic model verification technology is being compared to existing tools according to the following criteria:

- Considering verification as a search through reachable behaviors, the new tool *must not* find any behavior not actually reachable in the system. To make sure this is the case, we are comparing the random search tool's performance with that of the SMV model checker on models of a large flight guidance system (the SMV portion has already been done at the University of Minnesota).

[1] Because this document was delivered behind schedule, it contains information up-to-date 3/03, and there is significant overlap with the documents "Definition of Error-Seeding Methods for Modeling Languages" and "Definition of Validation Methods for Modeling Languages" delivered fall 2002.

- Again considering verification as a search through reachable behaviors, the new tool *should* find as much as possible of reachable behaviors of interest—that is, whatever behaviors influence the truth or falsehood of properties specified for the model.
- The new tool should work as fast as possible, and must require no more than approximately the same amount of time required by existing tools.
- The new tool should use significantly less memory than existing tools.

We have carefully checked all these criteria with randomly generated models over a large size range, for our tool, the SMV model checker, and the SPIN model checker (running in normal complete search mode and in a memory-saving approximation mode). Results from that experiment are available in "Lurch: a Lightweight Alternative to Model Checking" (lurch.pdf). We are now continuing to evaluate these criteria for flight guidance system models written in the UMN modeling framework.